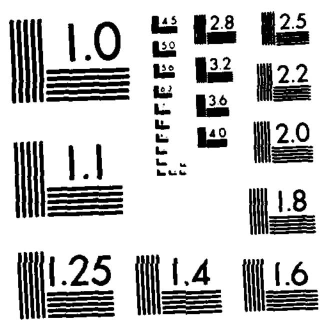
AD-A126 738 IMPLEMENTATION OF A COMPUTER LOCAL AREA NETWORK(U) 1/1
SCHOOL OF AEROSPACE MEDICINE BROOKS AFB TX
R A BALUSEK ET AL. FEB 83 SAM-TR-83-4

UNCLASSIFIED F/G 5/1 NL

END
DATE
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Report USAFSAM-TR-83-4

IMPLEMENTATION OF A COMPUTER LOCAL AREA NETWORK

Robert A. Balusek, M.S.
Bruce R. Montague, B.S.
Douglas H. Threatt, M.S.

February 1983

Final Report for Period 1 August 1978 - 31 May 1982

Approved for public release; distribution unlimited.

USAF SCHOOL OF AEROSPACE MEDICINE
Aerospace Medical Division (AFSC)
Brooks Air Force Base, Texas 78235

NOTICES

This final report was submitted by personnel of the System Operations Function, Computer Systems Branch, Data Sciences Division, USAF School of Aerospace Medicine, Aerospace Medical Division, AFSC, Brooks Air Force Base, Texas, under job order 7930-15-09.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder or any other person or corporation; or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Office of Public Affairs has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This report has been reviewed and is approved for publication.

Robert A. Balusek

ROBERT A. BALUSEK, M.S.
Project Scientist

Darwell E. Stowe

DARWELL E. STOWE, B.S.
Supervisor

Roy L. DeHart

ROY L. DEHART
Colonel, USAF, MC
Commander

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER USAFSAM-TR-83-4	2. GOVT ACCESSION NO. AD-A126738	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) IMPLEMENTATION OF A COMPUTER LOCAL AREA NETWORK		5. TYPE OF REPORT & PERIOD COVERED Final Report 1 Aug 1978 - 31 May 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert A. Balusek, M.S. Bruce R. Montague, B.S. Douglas H. Threatt, M.S.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS USAF School of Aerospace Medicine (BRS) Aerospace Medical Division (AFSC) Brooks Air Force Base, Texas 78235		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62202F 7930-15-09
11. CONTROLLING OFFICE NAME AND ADDRESS USAF School of Aerospace Medicine (BRS) Aerospace Medical Division (AFSC) Brooks Air Force Base, Texas 78235		12. REPORT DATE February 1983
		13. NUMBER OF PAGES 28
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Local Area Network
Computers
Communications
PDP-11
VAX-11
Distributed Data Processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper summarizes over three years of experience in implementing a computer local area network. A detailed description of the evolution of this network is followed by a list of lessons learned and conclusions drawn concerning future directions. The implemented network was conservative and classical in many respects. This is a technical paper written using a reflective and detailed style in the belief that some of the lessons of this experience may be of value to other implementors responsible for the nuts and bolts level of getting the job done.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

The authors are grateful to the people at the USAF School of Aerospace Medicine, Brooks AFB, Texas, who contributed suggestions and criticisms during the implementation of this computer network. Special thanks are due to Mr. Edward J. Engelken for his assistance in developing the master communication plan, and to Mr. Ronald Stone and Mr. Richard A. Leal for their technical efforts in implementing and maintaining the communications equipment. The authors are particularly appreciative of the computer operators in the System Operations Function who exhibited patience and interest during the many changes made during this period.

IMPLEMENTATION OF A COMPUTER LOCAL AREA NETWORK

INTRODUCTION

In the summer of 1978, the USAF School of Aerospace Medicine (USAFSAM) had some 20 small laboratory computers performing data acquisition and a Remote Job Entry system connected to an IBM 360/65 located at the San Antonio Data Services Center (SADSC), an Air Force Time Sharing Computer Service. Because of incompatible processors and storage media, transferring lab data to the 360 often took weeks. No school-wide interactive computer support was provided researchers; a significant application backlog existed; and strong user demand was expressed for a modern research support environment. These factors and others had been identified by a number of internal and external studies, resulting in a proposal for a local computer network that would link laboratory machines with a network host containing substantial memory, disk storage, and communication facilities. The USAFSAM physical plant consists of some 10 major buildings scattered over a number of miles, with most buildings being clustered together in one central area. The USAFSAM Data Sciences Division was charged with developing the network, and the System Operations Function was responsible for operating the network and maintaining and developing network software.

In the spring of 1982, the resulting network consisted of 21 laboratory data acquisition computer systems ranging from Digital Equipment Corp. MINCs to a PDP-11/45. Currently, 8 of these PDP-11s are capable of being DECNETed to the central network site. Four of the lab systems use the RSX-11M operating system; the others all use RT-11. The central site contains a large PDP-11/70, a VAX-11/780, and a PDP-11/34. The 11/70 uses the RSX-11M+ operating system and has 1.4Mb of memory, 402Mb of disk storage, and 8 DMC-11 DECNET communications devices. The 11/70 can use HASP to communicate with the SADSC IBM 4341, that replaced an IBM 360/65 in 1980, the AFHRL UNIVAC 1100/81, and CDC's Cybernet network. The VAX-11/780 uses the VMS operating system and has 4Mb of memory and 906Mb of disk storage.

DEVELOPMENT

The 11/70 was installed in late August of 1978, and the RSX-11M V3.1 operating system was installed in early September. RSX was chosen because it supported DECNET and

at least two vendors' HASP packages would run on it. RSX was also felt to be a sufficient real-time operating system for supporting a communications network. A version of FORTRAN, F4P, and a Basic compiler, BP2, were the supported application languages. The RMS-11K file system, which supported a powerful indexed file capability, and FCS, an older file system, were also provided. All editing was performed with the EDT editor. Other editors such as TECO and EDI were not used because TECO was considered too high in overhead and EDI was too primitive for our experienced user community to accept. Thirty-two terminals were connected to the 32 ports available on the 11/70 via a manual communications patch panel which also contained 4 modems providing dial-in support. Of the 32 terminals, 6 were VT55s and 26 were VT52s. Later, 2 Tektronix 4010 graphics terminals were added.

At this point, we were all learning. We started out with the standard RSX system. For some 3 months, only System Operations and key Data Sciences personnel used the system, after which week-long classes were held to train all Data Sciences personnel. During this period we noted some deficiencies in RSX: lack of a print spooler, and almost a total lack of accounting capability. Higher management considered accounting essential, so a version of the KMS Fusion accounting system was installed. This system, written in macro, was distributed on the Fall 78 Decus tape and required a number of modifications to MCR and to RSX. Additionally, this system altered the DEC account file, rather then keeping separate records, and a number of features desired, such as terminal tracking, were not included in the KMS package. Management wanted a mechanism for tracking terminal locations and the amount of time that a terminal was actually used in order to optimize terminal use and to prevent the problem of sophisticated lab users moving terminals without Data Science's knowledge.

A terminal tracking system that prompted the user for his terminal tracking number was incorporated in the login procedure. The number was attached to all terminals (a catchall number could be used for terminals dialing in). The assumed location of the indicated terminal was then displayed, and the user could confirm or alter this information. Location, phone number, and responsible individual were all tracked. Although the user's response could not be enforced or verified, USAFSAM does have a responsible user community and the system worked reasonably well.

The actual mechanism used to modify the login routine was to have the DEC program HELLO.MAC spawn a secondary login program. This program was written in BP2 and supported a complex accounting system based on indexed files. This program also handled some chores, such as

printing the login file and executing the users login command file, that were normally handled by DEC's HELLO program.

The DEC BYE program was removed and installed under another name, and we then installed our own program called BYE. This was a BP2 program that obtained, transparently to the user, the information needed to complete accounting for the user's session and update the secondary account files. The last thing this program did was execute the real DEC BYE program.

This procedure, upon first analysis quite a kludge, worked out very well. It had the advantages of being written in a high-level language, supporting indexed files, and requiring a minimum of modifications to RSX. Additionally all the resources now existed to implement inter-user mail, a who utility to display users (there was no ability to display users by name under RSX V3.1), and to easily write accounting reports. Another mandatory capability provided by this system was the enforcement of the concept of one active terminal session per user. If a user attempted to sign on a second time, he was simply signed off by executing the real DEC BYE program.

During this time, the training of over 100 users throughout USAFSAM had been accomplished by the Consultation and Training Branch of the Data Sciences Division.

Two RM03 67Mb disks existed on the 11/70 with one pack being used for the system and the other pack being dedicated to users. It rapidly became apparent that a means for enforcing a disk quota scheme would be required. Although the 11/70 had been conceived of as a communications handler and data concentrator, it was the first truly interactive computer to which the user community had been exposed, and it rapidly became a "poor man's" timesharing system.

A number of hardware problems plagued the first year of operation. Intermittent memory problems were experienced with a third-party's add-on memory, and this memory was finally removed. Persistent difficulties with static electricity occurred. For instance, bumping the system console into the adjacent work table would crash the system. Numerous efforts to assure a common isolated ground and monitor line power with a Dranetz Power Analyzer ensued. This effort ultimately resulted in acquisition of a Topaz isolation transformer and employment of large capacitors as spike suppressers.

Additional user-oriented system software included an enhanced version of inter-user mail and a system that permitted users to activate global command files that were stored in one location. The user did not have to know where

the single copy of the command file existed, nor did users need to make separate and redundant copies of the file. A single installed privileged program executed the required MCR command to activate the desired command file.

The lack of a print spooler that would handle multiple printers became a serious problem when a second printer was acquired and placed in the classroom area that was evolving into a terminal room. A spooling system had to be written. A privileged FORTRAN program was installed as the DEC spooler. This program could obtain and process a number of options from the user, and it maintained a global common that mapped terminals to line printers. The FORTRAN program acted as a top-level monitor and intercepted Send/Receive (S/R) packets intended for the DEC spooler. The FORTRAN program then interrogated the global data base to determine what action was to be performed, and retransmitted S/R packets to a version of the DEC spooler that drove the respective printer. A version of the primitive DEC spooler was installed for each printer to be serviced.

The STAT-11 interactive statistics package was converted from RSTS/E to RSX-11 and significantly enhanced. This package was available through DECUS and required substantial reorganization and overlaying efforts before it could run under RSX.

Pool space, the executive scratch work area, was already becoming an acute problem, especially when attempting to use DECNET. Without DECNET installed, 20 to 24 users on the system would exhaust pool, with the result that the system would freeze. A loading experiment was performed that detected a fatal bug in the RSX 3.1 task shuffler and illustrated that the system could not realistically support more than 12 to 14 users performing program development. It also illustrated the drastic loss of performance that resulted when memory became full and the system started swapping tasks in and out of memory. Users continued to write ambitious programs and to run into the 64Kb address space limitation of the 11/70's 16-bit architecture. It was becoming obvious that a machine on the order of a VAX was required.

Although it was not apparent at the time, some extremely significant system work was performed under the guise of a computer science graduate project. The project involved writing an interactive assembler for the PDP-11. It should be understood that Data Sciences had a long history of encouraging professional and academic research on their equipment, as long as idle machine time was used and the work was not performed during regular working hours. As a result of this effort, a complete set of macro subroutines became available that enabled BP2, a sophisticated compiler Basic, to be used for serious systems programming. As

almost all system utility programs developed since then have made use of these routines, the long-term impact of this work has been enormous.

As part of the same effort a preprocessor was written for BP2. This preprocessor, BPR, made Basic a worthwhile high-level language. Meaningful mnemonic labels were supported and Basic's percent sign soup integer data typing convention was replaced by the FORTRAN convention that variables starting with letters in the range I to N are of integer data type. This preprocessor rapidly became the predominant method used in writing application code, with over 87,000 lines of BPR source code existing as of this writing. It is worth emphasizing that in the normal operational or production environment such applied research might have been difficult to justify.

Once system programming tools were available, a number of projects were initiated. Work began on routines to read foreign magtapes, especially those produced by the SADSC IBM 360. Plotting support was provided, at first via a Calcomp 565. The DEC Plot-11 software drove this plotter directly online. Because of the heavy I/O burden caused by the online plotting, a routine to convert the 565 Calcomp plot output to a magtape capable of driving a CALCOMP 765 plotter was written (this involved converting from incremental graphics instructions to coordinate oriented plotting instructions). This same technique was also used to generate plots produced on the SADSC IBM computer system and sent by RJE to the 11/70.

With the advent of BPR, the use of BP2 expanded rapidly and RMS-11K indexed files became very popular. A variety of indexed file applications, mostly oriented towards data entry and forms retrieval, were developed. The system rapidly became saturated with such application programs.

The second major development was the establishment of the RJE link to the SADSC IBM 360. A number of alternative packages were examined under contract by Kentron International and the package recommended, HASP+ from Datanex Inc., was purchased. This package was first brought up on the system development 11/34 and worked quite well. Much of this package was written in FORTRAN in a fashion that facilitated user modifications. One of the outstanding features of this package was that any user at any terminal could submit an RJE job without operator intervention. This feature did not function at first, so a cover function was written which would output the file transfer request to the operator's console, after which the operator would manually transfer the jobstream. Operator utilities to display the status of HASP output files were written, and RJE plotting support was provided via a number of 11/70 utilities. Utilities were written to input data from the card reader

and to eliminate trailing blanks in files. An EBCDIC to ASCII translation routine was needed for files that were moved from the 360 to the 11/70 via punching Hasp output to the 11/70. After about two weeks of live checkout, the IBM System/3 RJE was replaced as the prime means of submitting jobs.

A data entry validation and forms management system, Series-IV by Informatics, was evaluated in a beta test site mode. Although there was considerable user interest in this package, and the package was good in and of itself, it could not be successfully integrated with the timesharing environment existing on the 11/70. The company's interest in their product was not strong, and thus recommended fixes were not vigorously pursued. The product was later dropped by the company.

At the end of the first year about 200 user accounts existed and during any one month perhaps 100 people would use the system. All varieties of users and skill levels existed. Over 60 terminals were now in use and 2 additional disks had been added, for a total of 4. Moving user's accounts around to support changes in disks was a nontrivial task, and serious disk space problems were occurring. The pool space problem was by now getting desperate. A DECUS Pool monitor program that warned of critically low pool had helped considerably, but was no solution. It was not uncommon to run out of pool 2 to 4 times a day and to run out of disk space, despite operations best efforts, once or twice a week. During this time a number of experienced users complained that they had never seen a timesharing system as bad as this. We usually agreed. Another daily operational problem was caused by the contention of some 60 terminals for 32 11/70 ports. Users had to call operations, and the operators would manually patch the desired line into a port.

When RSX-11 V3.2 became available, things improved considerably, as RSX 3.2 was more stable than 3.1. It had a print spooler and a batch queue, and the batch queue was used to replace the intolerable taskbuild times (on the order of 20 to 30 minutes) being experienced by RMS users. Magtape work continued. IBM Standard Label tapes could now be created and read on the 11/70, and unlabeled tapes could be produced in EBCDIC or ASCII. The Tektronix IGL package and Easygraph were installed, providing interactive graphics on a number of existing Tektronix 4010's and 4015's. The computer room electrical service was rewired in anticipation of necessary expansion.

A good disk space utilization scheme was implemented. In retrospect, it was probably this facility more than any other that in the end allowed us to operate RSX-11 as a timesharing system. The disk quota system was based on a

number stored in the user's secondary account record, indicating the maximum amount of disk space that the given user could use. With the macro system programming subroutines that were now available our BPR login program could be modified to process the users' file directory and the disk index file to determine the amount of disk space being consumed by the user at login. If the user was using more than his login quota, he was warned to reduce his disk space during the current session and a bit was set in the secondary account file to indicate that this user was in a period of grace. When the user attempted to logoff, he was again warned of this condition. If the user did not reduce his disk space, the next time he tried to log on, the warning bit would indicate that this user had been given his one session of grace and the user would be logged off. The only recourse for the user was to then contact System Operations. If the user attempted to sign on and it was found that his disk space was below his login quota, the bit would be cleared and all would be well.

One of the operational problems that worsened as user consumption of disk space increased was the down time caused by backups. Backups consumed a large amount of time, often nearly as much as machine down time. The DSC disk backup utility was replaced by the faster BRU utility, which used less tape than DSC, but which, upon first release, would on occasion corrupt indexed files. This resulted in a number of utilities to restore the contents of such files and to zap any block in a file and any physical block on disk. A number of corrupt disks (for instance, disks with overwritten home blocks) were recovered using these utilities.

The BMDP statistical package was installed on the 11/70 at this time. System use was by now split in half between those using the 11/70 for RJE support and those doing production work on the 11/70 itself. The Datatrieve query language began to be used for some real applications. Two more RM03's brought the disk total to six, with 3 being user packs, 1 the system pack, 1 a system support pack, and the last being used as a mountable pack.

Replacing the manual patch panel with a Digital Communications Associates (DCA) communications front-end provided a workable network for the first time. The DCA front-end provided facilities for users to establish their own connections, and it also permitted the use of concentrators in areas containing large numbers of terminals. The DCA was unique in that concentrators that performed statistical multiplexing were provided which actually plugged directly into the PDP-11 Unibus and could emulate many DEC DZ-11 communication devices on one board. The DCA was reasonably reliable and worked well, although it was again necessary to modify the login/out program. This

time, the problem was to control the relationship between the virtual connections, established by the user on the DCA, and the RSX logical terminal session. The user could abort his DCA established virtual circuit without logging off and would thus be unable to log on again because of the single/session philosophy. However, the next user for which the DCA established a virtual connection would like as not find himself in the midst of the first user's account! Likewise, when a user signed off, the BYE program had to somehow indicate to the DCA that the DCA was to disconnect the virtual circuit. This was more difficult than first appears because the DZ-11 is rather primitive and the RSX terminal driver did not provide a ready mechanism for controlling Data Terminal Ready. The RSX terminal driver could not even be told that the terminals were remote and thus required modem support, because only one remote baud rate could be supported on the DZ. We had decided to run all the terminals communicating directly with the DCA (using our own line drivers) at 1200 baud, but had to support real remote terminals dialing in at 300 baud. In the end a global I/O common was established that allowed the BYE program (and an initialization program) to simply access the DZ registers when they wanted to control the DCA, bypassing the RSX terminal driver completely. The DCA also permitted users on network terminals to access the AFHRL Univac 1108.

At this time, two superimposed networks were evolving. The DCA provided an asynchronous terminal-oriented network, while DECNET provided a synchronous file transfer-oriented capability. DECNET/RT was laboriously patched by hand and made to run between the 11/34 and the 11/70, but there appeared to be an educational problem in locating users. It had been a design objective from the start to have one of every peripheral used in USAFSAM on the 11/70. In addition, lab machines had to be guaranteed to run stand-alone in support of an experiment, so they all had some form of mass-storage. Thus, lab-support programmers tended to transfer data by carrying their tapes or disk packs under their arm. Since this method provided a very high effective baud rate, DECNET was a slow starter. Also, because DECNET used considerable pool space, its use was discouraged until RSX-11M+ was introduced.

The next phase of development was the acquisition of the VAX-11/780 for use as a network co-host. The installation of the VAX followed the same pattern used for the 11/70. The machine was used internally by System Operations, after which it was used for 5 or 6 months by key Data Sciences personnel. At first only 2 RM03 disks existed, and it was felt that more disk space was required before the machine could be effective. During this period, however, system software that provided the same functionality as that existing on the 11/70 was written. Control of the DCA during login/out was provided.

Enforcement of the single session/user concept was implemented. A DECUS accounting package was installed and deemed to be quite adequate. The magtape utilities, which had evolved by this time into a sophisticated utility package, were installed on the VAX. The PSTAT statistics package was installed.

At first users moved to the VAX because of the large address space, but they rapidly learned to appreciate the dramatic improvement offered by the VAX program development cycle, and soon the VAX was almost always the first machine considered for program development purposes. Users also began to run huge modeling programs that would not be feasible on a "fee-for-service" machine.

To experiment with DECNET, a permanent line was established between the 11/70 and the VAX. It was this line that made DECNET an overnight success. Users could access their 11/70 files with the VAX COPY command and thus perform all their own 11/70 to VAX file conversion. Users tended to do RSX program development and debug work on the VAX, and then move the finished application back to the 11/70. The 11/70 to VAX DECNET link rapidly became a heavily used resource and users would notice if this line was down in a matter of minutes. Additionally, DECNET was used by users to print on the 11/70 printer available in the main terminal room.

All during the development of the central facilities, a timely addition of various PDP-11s was being undertaken in the laboratories. Each new system required proper site preparation and installation of both hardware and software. After the initial system installation was completed, continued support was necessary. New releases of the RT-11 and RSX-11M operating systems were distributed and assistance provided to the application programmers using the lab systems. Some labs wanted process control, some wanted a data recorder, while others wanted to collect data and process it on the lab computer in real time. Few experienced application programmers were available to write real-time programs and interface hardware. In many cases, detailed knowledge of the operating system was necessary to accomplish the task so the task was assigned to the system personnel staff. Thus, a certain amount of time had to be spent in more direct lab support activity. One lab converted from RT to RSX, and System Operations performed the system work on this lab machine. This was the first lab machine to make real use of DECNET. Graphics devices such as the VT-11 and Ramtek and Grinnel graphics processors were supported. The system development 11/34 was used for systems work outside the lab environment, being notably useful when particularly difficult problems arose supporting the first release of the VT-11 software under RSX-11. All RSX sysgens for 11/34's were performed online on the 11/70.

This was possible because of the large assortment of peripherals available on the 11/70. The 11/34 was also used to print IBM output from SADSC offline. The utility that did this was a simple addition to the list of the ever-growing magtape utilities. The implementation of DECNET-RT on eight of the lab computers involved considerable effort. Many errors had to be corrected in the code as distributed by DEC. Eventually, we became a test site for the next release of DECNET-RT. This enabled us to finally complete the installation of DECNET-RT on the lab computers.

With the use of the VAX, it became much easier to modify HASP+. We had continued to receive upgrades of this product, and a number of utilities and HASP modifications had been made in an attempt to develop clean user display and postprocessing capabilities. When it was desired to explore the use of CDC's Cybernet, a version of the HASP+ code with different naming conventions was established. Thus, this copy of the HASP+ package was installed on the 11/70 without conflicting with the copy that communicated with SADSC. Unfortunately, only one device on the 11/70 could be used by the HASP+ package, so we had to patch this device between SADSC and the dial-up line to Cybernet. The use of the VAX SEARCH command made it relatively easy to modify HASP+ to support yet a third version when the AFHRL Univac acquired HASP support. Unfortunately, operators are still patching the single communications device manually upon demand. This has not yet become an extreme operational burden. Since the VAX and 11/70 communicate via DECNET and the VAX is rapidly becoming the central machine, a HASP interface was provided on the VAX. This system communicates with the SADSC 11/70 HASP system via a modified HASP+ module that is also installed as a DECNET-III known object. Recently, automated logging hooks have been placed in HASP+. This system will permit generating automatic reports of RJE activity.

The procurement of RSX-11M+ considerably increased the usability of the 11/70, and the DECNET network originally designed could now be built. Previous to M+, pool space problems had precluded the heavy use of DECNET. M+ and VMS together provided impressive DECNET Phase III capabilities, such as virtual terminal support and route-through. Route through permitted any lab system to access the VAX via the 11/70 without any need for additional hardware. While M+ was the most reliable version of RSX yet and could realistically support 20 to 24 users, we still experienced pool space problems, albeit at a much reduced rate. We had a very I/O intensive load involving heavy indexed file and Datatrieve activity and 8 DECNET communication devices. By this time, we had moved some of the smaller disks off the 11/70 and onto the 11/34 after discovering that there were some configurations of disks that would not coexist. Pool

space problems were finally almost totally eliminated by removing all M+ device error logging. While we have experienced a number of problems caused by the inability to detect devices going bad, this strategy has worked. We are hopeful that M+ V2.0 will permit us to reenable device error logging. The pool problem has been additionally ameliorated by the shift of user load to the VAX.

Among the more important DECNET support utilities is a routine to read RT unformatted FORTRAN files. While DECNET is able to convert among known file formats, RT FORTRAN uses an internal format for unformatted direct access files that the file system knows nothing about. Since lab data is often recorded in unformatted files, converting these files to RSX or VAX unformatted files is a necessary capability. This routine is yet another special magtape data conversion program. Indeed, a number of special routines that directly process various non-DEC and non-FILES-11 output tapes from specific lab machines are now supported.

A number of other operational functions that occurred during the implementation of this network include assisting in the conversion from MVT to MVS at SADSC, the maintenance of JCL procedures at SADSC, the installation of SAS 79.1 at SADSC, performing backups and creating library listings of SADSC disks used by USAFSAM users, relinking load modules stored as library members at SADSC and performing work to support the use of APT (a language for numerical machine control) at SADSC. A number of 11/70 utilities were written over an extended period to provide APT postprocessing capabilities.

OPERATIONS NOTES

The VAX-11/780 and PDP-11/70 are operated 7 days a week, 24 hours a day. Weekly backups are performed on Friday morning. These backups take 4 to 5 hours and usually start at 0600 in the morning. All sysgens are performed online. Critical parts of a system installation that require dedicated use of the machine are performed after hours or over the weekend, when usage is low.

CONCLUSIONS

The following is a listing of the lessons learned while implementing the network. Some of the ideas are based on observations while others are based on facts.

Network Notes

1. A local area network in a research environment need not have a static network structure. Diagrams of rings, stars, and trees often have little relation to reality, as network topology can change on demand. A patching system that can switch the configuration of synchronous interprocessor communication links is very useful.

2. The n-squared problem is a critical problem faced by network designers and integrators. The number of interrelationships goes up as the square of the number of systems. You simply cannot make dissimilar systems, optimized for different modes of operation, look like part of a single homogeneous network after the fact. Since all systems evolve, some mechanism for simultaneously evolving the interconnections and support utilities is required.

3. Adding point functionality to a network by adding special enhancements (making the network more heterogeneous) decreases the overall functionality and flexibility of the network. For instance, when all the terminals on the system were VT-52's, it was possible to clear the screen and do some nice things with cursor control on login and logout. As more terminals of every nature were added, it was no longer possible to support this capability.

4. A front-end processor is the only realistic way to provide a network involving different manufacturers' hardware. A significant benefit of the front-end is that all dial-in modems can be connected to it, thus permitting users to use one phone number to access all systems from their home or remote locations.

5. The use of asynchronous line printers that are local to user's terminals tends to destroy the statistical terminal session properties that statistical multiplexors depend upon.

6. It is essential for program development to have some means for the programmer to rapidly obtain local hard copy output.

7. If a front-end is used to establish connections between dissimilar systems, a means of maintaining the equivalence between virtual circuits and logical terminal sessions is required. If users can accidentally break their virtual circuits without terminating their logical terminal session, the most sophisticated security software will be useless, as users establishing new virtual connections to host machines will find themselves falling into the midst of

other users terminal sessions. Thus, tight coupling must exist between the front-end, which is maintaining virtual circuits, and the user session control software on the host machine. Likewise, software on the host that terminates the user's logical terminal session must trigger the front-end to break the virtual circuit or else the front-end will become tied down with a large number of phantom virtual circuit connections.

8. The only significant use of DECNET so far has been for file transfers. However, extensive use of this capability is made, particularly between the 2 timesharing hosts. Remote terminal capability is, for the most part, pre-empted by the use of the front-end to handle this function.

9. The front-end processor should have a fail-safe backup capability so that all or part of it can be bypassed if it should fail.

Conversions

1. Users are very reluctant to change systems. Improvements often make obsolete what has been a large part of their lives.

2. A good interactive editor is perhaps the most important item of software to make available to users. Such an editor will do more to encourage people to use the system than any other effort. Such an editor should support a screen or window mode. When a screen mode is used, you see less errors of the form where 2 lines are accidentally duplicated or exchanged.

3. Users often consider ease of use to be directly related to the number of characters they are required to type before they find themselves in the functionality calculator of their choice.

4. Help files are not as helpful to the naive users who really need them as they are to sophisticated users who already know what to ask. The people who really need them are those who type poorly and are reluctant to experiment with help probes.

5. Foreign magtape support is essential. Not only will the operations staff be expected to handle tapes coming in from research sites all over the world, but in many cases you can significantly reduce I/O costs. Line transmission costs are often high, and tapes can usually be made with block sizes larger than those used in

transmission schemes. Thus, a tape will often cost only a few thousand I/O's. Since users of communication software tend to move files by listing the file, which often costs an I/O per line, tremendous savings of time and money can result from the effective use of tapes.

6. We received a number of suggestions to start over with UNIX. UNIX is a development system, not a friendly production system for naive users. Also UNIX lacked adequate support when this network was designed, and we did not have the manpower to do all the things ourselves that would have been required for us to achieve all of our local network goals. We could *not* afford a one-of-a-kind system, because we could not be without a guaranteed avenue of growth.

7. It takes a shop a year to become comfortable on a new system, with a core of experienced users. Up to that time providing elementary assistance will be an ongoing responsibility of operations personnel.

8. During an upgrade, completely reevaluate the need for all systems, rather then simply convert the systems as users are accustomed to do. Users try to see new systems as better versions of the old system, rather than as different systems that may provide radically different means of accomplishing similar goals.

Techniques

1. A good communications *line monitor* datascope that can easily be connected to any user's line is a must for effective operation of a sophisticated local area network. The datascope can be used to spy on any user's line and often has overtones of big-brotherism, but it is essential in problem isolation and in providing user assistance. It is not uncommon to have a frustrated user call operations and request help. Without a datascope to observe what is happening, extricating the user can be very difficult. With the datascope, one can watch the user, see the error messages, and tell the user exactly what to do. In complicated cases, we have found it useful to make use of a facility for forcing commands to the user's terminal, which permits us to simply take over and proceed to demonstrate to the user what to do. This interactive guide mode can be invaluable. A complicated matter is explained with minimal effort by simply showing the user. If the user is in a remote location (across the nation), this facility may be the only way to effectively teach a user those little points that he missed in the documentation.

We would also vote, if we could, for a similar facility to be included as a privileged utility in timesharing operating systems. Such a utility is not without precedence. The CDC PLATO system has such a capability, for instance.

2. A subject is difficult if the subject material cannot be seen. For instance, communications software is considered difficult because it is so ethereal. This need not be true. A good datascope can make the coding of communications software as simple as writing a report. We are consistently surprised to find out that items such as magtape formats and communications schemes such as HASP and DECNET are rather simple once you discover how to see what it is they do.

3. A compiler Basic is an effective application language if the user does not have to use line numbers and the language supports a complete set of file, structure, error handling, and other high-level PL/1 like features. DEC's VAX-11 Basic and BP2 are such compilers.

4. A research environment demands a language that supports the rapid development of code. The reliability or formal accuracy of the program may not be as important as rapidly achieving one objective with a particular set of data in a minimum amount of time. Programs are often run only once, which results in the writing of a lot of throw-away code. Since many of these programs are self-verifying in that the researcher will be able to tell if his results are correct, this approach is not as dangerous as first appears. Research programmers often program by modifying or mutating existing programs. A language such as Basic, with some necessary enhancements, can be adequate for much of this work.

5. The two most successful application support packages were probably the RMS-11K indexed file system, and FMS-11, a forms management system. With the availability of RMS-11K, users found themselves with a powerful indexed file capability. RMS-11K provides for self maintaining indexed files that were significantly easier to use than other indexed file schemes that users had been exposed to on other machines. It became common to see files in which the majority of the fields in a record were indexed. FMS provided a rapid means of developing sophisticated forms entry and display software.

6. A good interactive editor encourages the programming at the keyboard syndrome. This is not bad as long as the design effort also takes place at the keyboard. With a good interactive editor everything that can be done at a desk can be done at a terminal, and all structured

design and specifications work can be captured immediately in machine readable form. All documents relating to a program should be machine readable and should be included with the code and data files in a single directory so that the entire software system can easily be moved to another machine. What can be done on paper that cannot be done online if the user can type and is skilled with a truely versatile editor?

7. Avoid modifying the executive. Do not fight the next release. Consider the likely lifetime of any software that you are tempted to write.

8. Unless data is physically overwritten, it is almost always possible to recover data from corrupted disks or files. Investing in understanding the physical on-disk structure and writing utilities to do physical I/O against disks may be well worth the effort if you ever have to recover such lost data in an emergency. In one case data from a file was recovered by sequentially reading an entire disk and recovering every block that contained a pattern known to be in the records of interest! Do this sort of work, if possible, in a high-level language so your work will have some lifetime. Such schemes also make it possible to violate any software security schemes protecting data or files.

Observations

1. One recurring limitation to system growth was disk space. Managing disk space is an essential capability. The implementation of disk quotas under RSX made RSX a usable timesharing system.

2. RSX-11M+ is an adequate timesharing system for about 20 users. However, the program development cycle for programs that must link with RMS is too long.

3. We appear to have evidence to support the conclusion that "significant improvements lead to reduction in the number of simultaneous users"[1], and also to agree with the conclusion in the same paper that "a handful of people consume most of the machine resources."

4. You can not reliably provide a standard user interface by putting software on top of operating systems that are just not compatible.

5. Most research users see a computer as a functionality calculator. They want the ability to run BMDP or FORTRAN or whatever. That's all they care about. Cute

command languages and JCL that can optimize machine use are all seen as a burden that system designers should have eliminated by making it all automatic. Even application programmers almost always use only a small subset of a computer's resources. For instance, they will not use a system service that does not have a cover function in the language with which they are familiar.

6. Nouns are a class of words that can easily be known but not necessarily understood. Since there is no weighting factor for the importance or scope of technical terms, especially those concerning software, much confusion arises. Anyone can put up a display that gives the appearance of providing this or that functionality, but one must understand what the internal logic really does. Generalities obscure meanings. Although educating decision makers is part of any technical job, it is rarely possible to do more than provide superficial understanding in a realistic amount of time. One result of this problem is that people will confuse low-level communications schemes (such as RS-232C, X-25 and Ethernet) with high-level networking protocols. Perhaps a little education is a dangerous thing, since people then know enough to know the vocabulary but may not understand the detail level. People may thus appear to know more than they do, which contributes to confusion.

7. The concept of Office Automation is not well understood at the levels of management required for organization-wide support of a single unifying concept. Office Automation is most often used as a reason for placing microcomputers in dispersed locations, but it is not realized that Office Automation is actually much more than that. Adequate consideration of the communications and software substrate that must integrate such systems is often overlooked. This misunderstanding may be one of the bigger problems facing data processing in the next few years.

8. A significant amount of work done on a research machine will be reports and papers, which can perhaps be seen as software aimed at people rather than machinery. As much as 20% of all user terminal time might realistically be placed in this category. This is the type of activity that causes the distinction between research computers and management support systems (which often include word processors) to blur.

9. A continual problem faced in the operation of a local area network arises when operations staff are not involved or consulted in matters of design concerning some ad-hoc project, but they are then expected to make it work. No one knows a system better than the people who put it together and made it work. The many

differences between vendors' hardware and software makes the integration of these vendors' equipment into the network difficult. It is important to have a unified master network plan.

10. Perhaps the fastest way to disseminate key information, techniques, and changes is to cultivate a small group of key or dominant users. These are the users that must make use of a feature if it is to succeed. Show these people what they gain by using a new capability and they will then educate the rest of the user community.

11. A weekly meeting provided a forum for communicating with members of the programming staff. Even when all the tools are in place, and literature is available, people will rarely learn new information by themselves until it is made obvious to them that the new techniques are worth their time. Interestingly enough, in the case of the VAX, the machine was available for almost half a year with almost no use until a handful of 30-minute classes were performed during the programmer's meetings. After that, VAX usage rose dramatically.

12. Special projects, especially interfacing ad-hoc devices, are a problem. These projects expand to consume all system programmers' time, to the exclusion of supporting the basic infrastructure of the network. High visibility or urgent projects will be given priority over the implementation of network capabilities and utilities, which can always be put off another month.

13. Temporary evaluation systems are almost always permanent unless there are gross difficulties. Very often this leads to the failure to make resource allocation decisions and the like up front.

14. Sixteen-bit systems are too small for full-functionality statistics packages. The memory addressing of large data arrays is very difficult to achieve, and overlaying the various program sections slows execution.

15. Users' experience grows rapidly with use of the system, but training and continued support to these users is a very large task since their problems become more difficult to diagnose.

Reliability

1. Because disk space was inevitably the critical network resource and backup procedures were a major factor contributing to down time, procedures for providing

transparent, automatic archival and data set migration would have been of immense benefit. Unfortunately, Files-11 only keeps track of the date of last write access, so it is impossible to determine the date of last read access. Backup procedures should be transparent to users and should not require user involvement, since typical users have enough problems without shouldering the responsibility for their own backups. This item is included under reliability since it significantly impacts the availability of the system.

2. Efforts to make a machine do more than what it is easily capable of doing are counterproductive. Much system programmer time will simply be wasted as improvements are made obsolete by later releases of the system.

3. Running all host machines around the clock not only is desirable, but also improves overall system reliability. We almost inevitably had increased down time after an extended system shutdown. We experienced no significant problems in running machines unattended at night over a 3-year period.

4. A sure way to cause reliability problems is to completely load down a machine with options. As the number of peripherals increases, we found that some peripherals simply could not be made to coexist on the same machine. Unfortunately our strategy of having one of every peripheral that existed in the labs on the network host caused us to encounter this problem on occasion. Another problem caused by a "fully blown" configuration concerns cabling. On the host 11/70 serious cabling problems have occurred on a number of occasions. Preventive maintenance, or any activity that requires the system cabinets to be opened, is a dangerous time because of the number of cables that are vulnerable to damage. Similar circumstances often surround lab systems that are also ambitiously configured. Such a situation tends to give users an incorrect feel for the reliability of computers. Once learned this is not a lesson that is easily unlearned. These are problems that manufacturers can, and should, eliminate.

5. A shop is only as reliable as its air conditioning and electrical power conditioning.

6. The entire terminal communication network is only as reliable as the hardware/software of the DCA front-end processor. Continued growth of the number of terminals and printers soon exceeded the capacity of the front-end processor, necessitating expansion. In addition, problems with run-away terminals required modifications

to the line drivers used to connect the terminals to the network.

Successes

The following is a listing of the successes experienced during the installation and expansion of the network:

1. The BPR preprocessor for DEC's compiler Basics.

2. The HASP+ RJE system.

3. The DECNET interface between the host machines.

4. The DECNET interfaces between the 11/70 and the lab computer systems.

5. The secondary accounting system for RSX.

6. RSX Mail.

7. Software to control the DCA front end.

8. The foreign magtape package. This is probably our most significant achievement in terms of adding systems capability. Development of expertise in this area has permitted us to process CDC mixed mode magnetic tapes (character, floating point, and integer data) and to read and write IBM Standard label magnetic tapes. A complete set of tape utilities enables us to map and obtain data off almost any tape, although tapes with formats internal to various operating systems often require special cleanup work.

Some Systems That Failed

It is illustrative to consider some systems that failed. Among the efforts that can be classified in this category are:

1. An attempt to write an emulator for the DEC VAX MAIL program that would run under RSX. We had already written a useful but simple mail program used under RSX. The new program was successfully completed, but it was unwieldy code and inefficient. A prime reason for this effort had been to support distribution lists. This package worked very similarly to VAX Mail in that it produced a separate copy of a mail item for each user on the list. While the VAX is normally fast enough for

this to work, the 11/70 was by this time heavily loaded, and such a package was inordinately slow. The alternative was to support an unwieldy user data base.

2. Converting HASP+ to support a DMA device. This effort was dropped when the vendor decided to attempt the same effort.

3. The use of the Series-IV data entry capability. This package was intended to run on a stand-alone system, and could not be adapted to run in a timesharing environment. The vendor eventually dropped the effort.

4. Supporting IBM CMC terminals, and reading IBM mag cards. While it was possible to drive CMC terminals as slave output devices, reliably supporting automatic input from these half duplex terminals in dump mag card mode required more control then the DZ-11 permitted.

5. A home-brew soft-fail temperature shutdown system. This system, based on an 8080, turned out to be overkill. A power trip on the thermostat was simple, safe and reliable.

6. Support of the DCA communications software via a PDP-8 cross assembler. This effort was dropped because the anticipated lifetime of such a support environment would not justify the cost of becoming proficient in maintaining it.

7. Early efforts to do significant graphics using the VT55 terminals. The graphics capabilities of the VT55 were too limited, since plots having more than two of the same y coordinates could not be processed.

8. The first release of DECNET-RT had many errors and had to be patched by hand before it would communicate with DECNET-11M on the 11/70.

9. Online digital plotting caused such an excessive I/O load on the system and operation problems that it was replaced by an offline method.

Some Systems That Died

1. Terminal independent cursor positioning and screen I/O subroutines were replaced by FMS-11.

2. Early accounting packages.

3. User utilities to examine the status of user's RJE jobs and allow users to scan their output. These never turned out to be as useful as anticipated. Since they were all basically postprocessors, they would require modification every time a change occurred in the MVT/MVS job listing format.

4. Converting Plot-11 graphics output to Calcomp format and converting Calcomp output to Tektronix output. Plot-11 graphics became obsolete with the procurement of Calcomp software. An APL package that converted Tektronix to Calcomp graphics had been written for the SADSC 360. Although work was begun on converting this system to a package for previewing Calcomp plots, little user interest developed and the effort was dropped.

5. Various schemes to support more pool under RSX.

6. Various online documentation schemes. These never gained widespread user acceptance.

7. Global naive user-oriented command files to guide users through typical program development cycles. Apparently there is no such thing as the typical naive user or typical program development cycle in a research environment.

8. Support of APT. We had been providing assistance (chiefly postprocessors to extract machine control data) for the use of APT at SADSC. After the conversion at SADSC to MVS, APT became too expensive for our users. An alternative APT was available on the AFHRL Univac (by this time an 1100/80), and our users could use the front-end to directly access this machine.

9. The RSX 3.1 print spooler died with RSX V3.1.

FUTURE NETWORK EXPANSION

The following is a listing of future directions and trends which will influence the continued implementation and expansion of the network:

The Wishlist

1. Every system utility and all programs should be callable as system services (predefined subroutines). This would permit application software, such as data base managers and statistics packages, to all use a common editor.

Users should be easily able to add their own routines to the core system.

2. Likewise only one form of every functional mode should exist. A single command level should exist. There should be one editor and one mechanism for moving data, whether the data be in files or in fields within those files. Currently, each system has its own version of a given functional capability. A monolithic man/machine interface with a single, comprehensive, command repertoire would be an immense productivity tool.

3. A user should see only one screen at any time. Programming languages should not be able to produce more inline code than will fit on a single screen. Instead of flat files, programs should be organized as a hierarchial "tree" of functional blocks, each of which will be constrained to fit on a single screen. Rather than scroll forward and backward, the user should exist in a 3-D space in which information that cannot fit on his screen is compressed into a named window. The user could thus still track and comprehend his entire system, seeing it as a central window linked into a tree of various smaller windows. The user should be able to zoom in and out of these windows, dropping into a subroutine or documentation window and then popping out to a higher level in the tree of components that constitutes the program. Windows of documentation and code should be intermingled.

4. It might be advantageous for hardware to contain a mechanism for indicating missing data or infinity. This would be a value that could be considered as zero (or perhaps any number) if a bit in the processor status word was enabled. If the bit was disabled, the value could be tested, setting a missing or valid flag.

5. A network should make data appear as a common set of file cabinets or as a common bulletin board. At the present time no good solution readily provides such capability across dissimilar machines.

6. It is not possible to reliably model a network with any tools available today. Somehow it is necessary to model the adaptability of a configuration rather than just the capabilities of a static configuration.

7. Often a tendency in the design of local networks is to solve the easy parts and not consider the difficult parts. No doubt some of this is due to the fact that users cannot really design local networks; they can only choose between systems in much the same manner that the buyer of a car can only design his car to the limited extent of choosing options. Since there is

always user pressure for progress, actions are sometimes taken without a serious attempt to integrate the overall effort. One of the more frustrating efforts that occurred during the final phases of this endeavor was to do an analysis of the requirements of a workable distributed office automation network and discover that there was no desire to answer some of the questions raised. In part this was due, no doubt, to the realization that no realistic answers existed. Impressing upon people that the software to support such efforts is often very primitive is difficult because users tend to consider computers as advanced and sophisticated machines.

Trends

1. Ultimately (5 years) we will have to support large numbers of users and visiting/contracting researchers who bring their own personal systems with them. Most researchers today have personal calculators and typewriters. Personal systems with computational capabilities equivalent to those of any central timesharing system we could afford will exist. Significant amounts of portable secondary storage will make it possible for researchers to do the bulk of their work on the computer that followed them through graduate school. This will boost productivity but present us with the problems of providing a common communications facility, implementing some sort of data library, and pose interesting software control and security problems. Conversion facilities such as those in the present foreign magtape package may assume even greater importance. We will still have to provide some monolithic system for computer illiterate users performing functions such as routine data entry and forms management.

2. Word processing strategies involving central pools assume that the bulk of their users will remain computer illiterates. This situation will no doubt be true for a number of years but can be anticipated to change. Once users learn a good editor, be it even on their personal machine, they will be reluctant to go back to longhand. Recent advances in radical keyboard design strategies may also significantly lower the number of users writing on paper in longhand.

3. Very high storage densities should be achievable with video-disk-like technology. Giga- and tera-bit write-once memories may require novel techniques for organizing secondary storage.

4. Many stand-alone terminals will adopt high resolution memory bit-mapped screens similar to the Xerox Star. Using this high resolution graphics terminal, users will be able to design their own character sets and will be able to use a graphics editor to simply draw special text on the screen.

5. System maintenance and updates will become automatic. The manufacturer will simply call the computer and perform the updates over the wire.

6. Local area networks will require high-speed interconnects, either fiber optic or coax. For systems that are truely of a distributed computer nature, baseband systems will probably predominate. Such systems are essentially expanded computer busses and thus will need to be able to support simultaneous multi-casting to all devices on the net/bus from any point on the net. Simultaneously monitoring and controlling a large number of devices is difficult if all the devices are communicating on different frequencies in a broadband fashion.

Long-Term Directions

1. Synoptic displays: as things become more sophisticated it will become necessary to be able to see the state of a machine or of software. In field service, operations, and applications programming environments, packages that make use of high resolution graphics and present the user with a diagram presenting an analog of the actual situation would be extremely helpful.

2. A single user environment: the distinction between operating systems, utilities, and programming languages should go! The user would find himself in one common environment similar to that provided by FORTH.

3. Central data facilities: computing consists of nothing more than changing, moving, and storing data. In a research environment in which many individuals work on the same data, that part of the system that serves as a central library will become the logical hub of the system.

4. Very high speed local networks: optical technologies may achieve terabit transmission speeds by the end of the decade. Television will go digital, which will allow the utilization of digital data compression and error correction techniques, and thus be an added reason

for networks to adopt baseband systems. Baseband systems are essentially digital, while broadband systems are analog. Digital information can be switched by software while analog information requires more expensive hardware.

5. Application generators: programmers are fundamentally no more than translators. We need tools that make the translation between the end-user's statement of his problem and the application program more automatic. These tools will belong to that class of Artificial Intelligence programs known as expert systems, with the domain of expertise being that of programming and of acting as a programmer's apprentice. Even if such systems only produce template programs, dramatic increases in programmer productivity might be achievable. Extensible programming systems that can support alternate programming environments should move programming closer to the problem domain. Efforts in natural language recognition may also contribute a great deal. It is important to recognize that exciting results might be possible with even incomplete natural language processing systems.

REFERENCES

1. Doherty, J., and R.P. Kelisky. Managing VM/CMS systems for user effectiveness. IBM System Journal, Vol 18, No. 1, 1979.

LME

8